

- **Engage** Engagement with various stakeholders to provide assurance with regard to an organization's readiness for disasters is supported by this practice.
- **Design and transition** Service continuity management ensures that products and services are designed and tested according to the organization's continuity requirements.

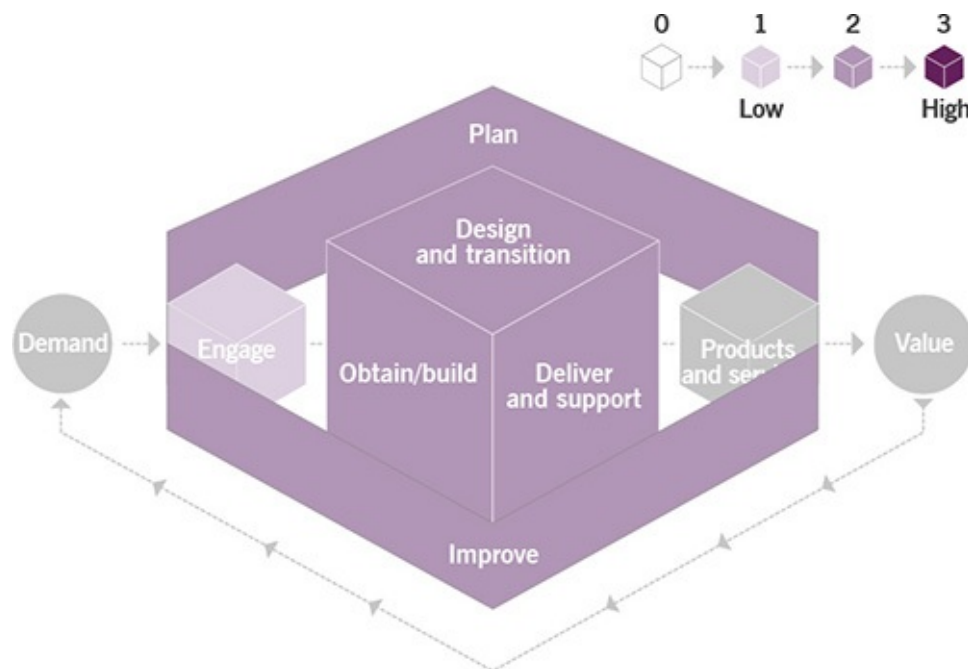


Figure 5.31 Heat map of the contribution of service continuity management to value chain activities

- **Obtain/build** Service continuity management ensures that continuity is built into the organization's services and components, and that procured components and services meet the organization's continuity requirements.
- **Deliver and support** Ongoing delivery, operations, and support are performed in accordance with continuity requirements and policies.

### 5.2.13 Service design



#### Key message

The purpose of the service design practice is to **design** products and services that are fit for purpose, fit for use, and that can be **delivered** by the organization and its ecosystem. This includes **planning** and **organizing** people,

partners and suppliers, information, communication, technology, and practices for new or changed products and services, and the interaction between the organization and its customers.

If products, services, or practices are not designed properly, they will not necessarily fulfil customer needs or facilitate value creation. If they evolve without proper architecture, interfaces or controls, they are less able to deliver the overall vision and needs of the organization and its internal and external customers.

Even when a product or service is well designed, delivering a solution that addresses the needs of both the organization and customer in a cost-effective and resilient way can be difficult. It is therefore important to consider iterative and incremental approaches to service design, which can ensure that products and services introduced to live operation can continually adapt in alignment with the evolving needs of the organization and its customers.

In the absence of formalized service design, products and services can be unduly expensive to run and prone to failure, resulting in resources being wasted and the product or service not being customer-centred or designed holistically. It is unlikely that any improvement programme will ever be able to achieve what proper design could have achieved in the first place. Without service design, cost-effective products and services that deliver what customers need and expect are extremely hard to achieve.

Service design practice should also ensure that the customer's journey from demand through to value realization is as pleasant and frictionless as it can be, and delivers the best customer outcome possible. This is achieved by focusing on customer experience (CX) and user experience (UX).

Adopting and implementing a service design practice focused on CX and UX will:

- result in customer-centred products and services that include stakeholders in design activities
- consider the entire environment of a product or service
- enable projects to estimate the cost, timing, resource requirement, and risks associated with service design more accurately
- result in higher volumes of successful change
- make design methods easier for people to adopt and follow
- enable service design assets to be shared and re-used across projects and services
- increase confidence that the new or changed product or service can be delivered to specification without unexpectedly affecting other products, services, or stakeholders

- ensure that new or changed products and services will be maintainable and cost-effective.

It is important that a holistic, results-driven approach to all aspects of service design is adopted, and that when changing or amending any of the individual elements of a service design, all other aspects are considered. It is for this reason that the coordination aspect of service design with the whole organization's SVS is essential. Designing and developing a new or changed product or service should not be done in isolation, but should consider the impact it will have on:

- other products and services
- all relevant parties, including customers and suppliers
- the existing architectures
- the required technology
- the service management practices
- the necessary measurements and metrics.

Consideration of these factors will not only ensure that the design addresses the functional elements of the service, but also that the management and operational requirements are regarded as a fundamental part of the design, and are not added as an afterthought.

Service design should also be used when the change being made to the product or service is its retirement. Unless the retirement of a product/service is carefully planned, it could cause unexpected negative effects on customers or the organization that might otherwise have been avoided.

Not every change to a product or service will require the same level of service design activity. Every change, no matter how small, will need some degree of design work, but the scale of the activity necessary to ensure success will vary greatly from one change type to another. Organizations must define what level of design activity is required for each category of change, and ensure that everyone within the organization is clear on these criteria.

Service design supports products and services that:

- are business- and customer-oriented, focused, and driven
- are cost-effective
- meet the information and physical security requirements of the organization and any external customers
- are flexible and adaptable, yet fit for purpose at the point of delivery
- can absorb an ever-increasing demand in the volume and speed of change
- meet increasing organizational and customer demands for continuous operation

- are managed and operated to an acceptable level of risk.

With many pressures on the organization, there can be a temptation to ‘cut corners’ on the coordination of practices and relevant parties for service design activities, or to ignore them completely. This should be avoided, as integration and coordination are essential to the overall quality of the products and services that are delivered.

### 5.2.13.1 Design thinking

---

Design thinking is a practical and human-centred approach that accelerates innovation. It is used by product and service designers as well as organizations to solve complex problems and find practical, creative solutions that meet the needs of the organization and its customers. It can be viewed as a complementary approach to Lean and Agile methodologies. Design thinking draws upon logic, imagination, intuition, and systems thinking to explore possibilities and to create desired outcomes that benefit customers.

Design thinking includes a series of activities:

- **Inspiration and empathy**, through direct observation of people and how they work or interact with products and services, as well as identifying how they might interact differently with other solutions.
- **Ideation**, which combines divergent and convergent thinking. Divergent thinking is the ability to offer different, unique, or variant ideas, while convergent thinking is the ability to find the preferred solution to a given problem. Divergent thinking ensures that many possible solutions are explored, and convergent thinking narrows these down to a final preferred solution.
- **Prototyping**, where these ideas are tested early, iterated, and refined. A prototype helps to gather feedback and improve an idea. Prototypes speed up the process of innovation by allowing service designers to better understand the strengths and weaknesses of new solutions.
- **Implementation**, where the concepts are brought to life. This should be coordinated with all relevant service management practices and other parties. Agile methodology can be employed to develop and implement the solution in an iterative way.
- **Evaluation** (in conjunction with other practices, including project management and release management) measures the actual performance of product or service implementation to ensure acceptance criteria are met, and to find any opportunities for improvement.

Design thinking is best applied by multi-disciplinary teams; because it balances the perspectives of customers, technology, the organization, partners, and suppliers, it is highly integrative, aligns well with the organization’s SVS, and can be a key enabler of digital transformation.

### 5.2.13.2 Customer and user experience

---

The CX and UX aspects of service design are essential to ensuring products and services deliver the desired value for customers and the organization. CX design is focused on managing every aspect of the complete CX, including time, quality, cost, reliability, and effectiveness. UX looks specifically at the ease of use of the product or service and how the customer interacts with it.

#### Lean user experience

Lean user experience (Lean UX) design is a **mindset, a culture, and a process** that **embraces** Lean–Agile methods. It **implements** functionality in minimum viable increments, and **determines** success by measuring **results** against an outcome **hypothesis**. Lean UX is **incredibly** useful when working on projects where Agile development methods are used. The core objective is to focus on **obtaining feedback** as early as possible so that it can be used to **make** quick decisions.

Typical questions for Lean UX might include: **Who** are the customers of this product/service and what will it be **used for?** **When** is it used and **under** what circumstances? What will be the **most important functionality?** What are the **biggest risks?**

There may be more than one answer to each question, which creates a greater number of assumptions than it might be practical to handle. The team will then prioritize these assumptions by the risks they represent to the organization and its customers.

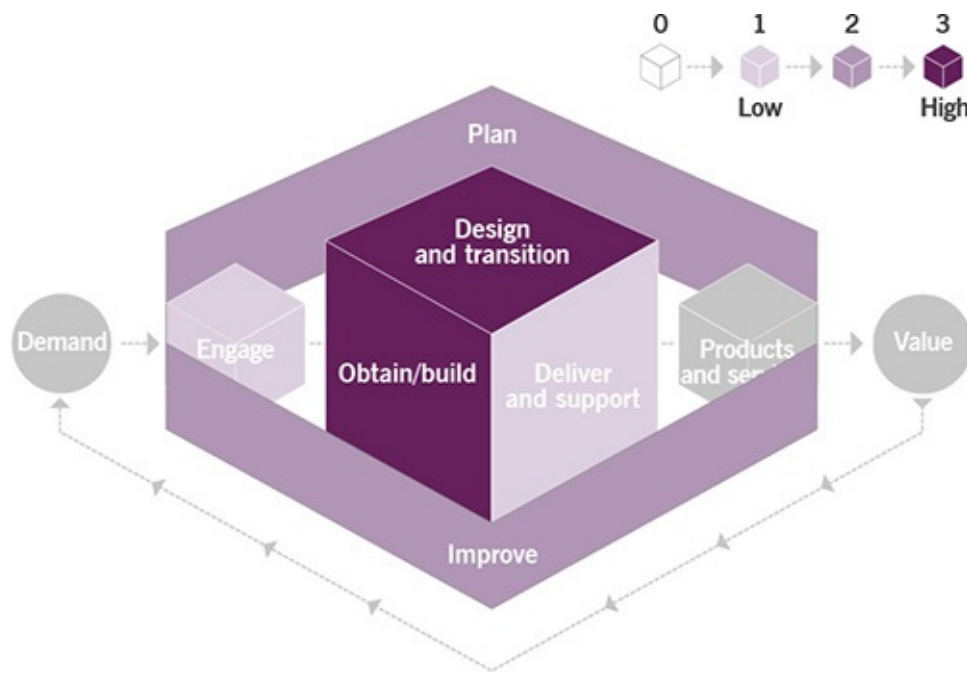


Figure 5.32 Heat map of the contribution of service design to value chain activities

Risk identification, assessment, and treatment are key requirements within all design activities; therefore risk management must be included as an integrated aspect of service design. This will ensure that the risks involved in the provision of products and services and the operation of practices, technology, and measurement methods are aligned with organizational risk and impact, because risk management is embedded within all design processes and activities.

Figure 5.32 shows the contribution of service design to the service value chain, with the practice being involved in all value chain activities:

- **Plan** The service design practice includes planning and organizing the people, partners and suppliers, information, communication, technology, and practices for new or changed products and services, and the interaction between the organization and its customers.
- **Improve** Service design can be used to improve an existing service as well as to create a new service from scratch. Services can be designed as a minimum viable service, deployed, and then iterated and improved to add further value based on feedback.
- **Engage** Service design incorporates CX and UX, which are quintessential examples of engagement.
- **Design and transition** The purpose of service design is to design products and services that are easy to use, desirable, and that can be delivered by the organization.
- **Obtain/build** Service design includes the identification of products, services, and service components that need to be obtained or built for the new or changed service.

- **Deliver and support** Service design manages the user's full journey, through operation, restoration, and maintenance of the service.

## 5.2.14 Service desk



### Key message

The purpose of the service desk practice is to **capture demand** for incident resolution and service requests. It should also be the **entry point** and **single point of contact** for the service provider with all of its users.

Service desks provide a clear path for users to report issues, queries, and requests, and have them acknowledged, classified, owned, and actioned. How this practice is managed and delivered may vary from a physical team of people on shift work to a distributed mix of people connected virtually, or automated technology and bots. The function and value remain the same, regardless of the model.

With increased automation and the gradual removal of technical debt, the focus of the service desk is to provide support for 'people and business' rather than simply technical issues. Service desks are increasingly being used to get various matters arranged, explained, and coordinated, rather than just to get broken technology fixed, and the service desk has become a vital part of any service operation.

A key point to be understood is that, no matter how efficient the service desk and its people are, there will always be issues that need escalation and underpinning support from other teams. Support and development teams need to work in close collaboration with the service desk to present and deliver a 'joined up' approach to users and customers.

The service desk may not need to be highly technical, although some are. However, even if the service desk is fairly simple, it still plays a vital role in the delivery of services, and must be actively supported by its peer groups. It is also essential to understand that the service desk has a major influence on user experience and how the service provider is perceived by the users.

Another key aspect of a good service desk is its practical understanding of the wider business context, the business processes, and the users. Service desks add value not simply through the transactional acts of, for example, incident logging, but also by understanding and acting on the business context of this action. The service desk

should be the empathetic and informed link between the service provider and its users.

With increased automation, AI, robotic process automation (RPA), and chatbots, service desks are moving to provide more self-service logging and resolution directly via online portals and mobile applications. The impact on service desks is reduced phone contact, less low-level work, and a greater ability to focus on excellent CX when personal contact is needed.

Service desks provide a variety of channels for access. These include:

- phone calls, which can include specialized technology, such as interactive voice response (IVR), conference calls, voice recognition, and others
- service portals and mobile applications, supported by service and request catalogues, and knowledge bases
- chat, through live chat and chatbots
- email for logging and updating, and for follow-up surveys and confirmations. Unstructured emails can be difficult to process, but emerging technologies based on AI and machine learning are starting to address this
- walk-in service desks are becoming more prevalent in some sectors, e.g. higher education, where there are high peaks of activity that demand physical presence
- text and social media messaging, which are useful for notifications in case of major incidents and for contacting specific stakeholder groups, but can also be used to allow users to request support
- public and corporate social media and discussion forums for contacting the service provider and for peer-to-peer support.

Some service desks have a limited support window where service cover is available (for example, 08.00–20.00, Monday–Friday). Staff are therefore expected to work in shift patterns to provide consistent support levels.

In some cases, the service desk is a tangible team, working in a single location. A centralized service desk requires supporting technologies, such as:

- intelligent telephony systems, incorporating computer-telephony integration, IVR, and automatic call distribution
- workflow systems for routing and escalation
- workforce management and resource planning systems
- a knowledge base
- call recording and quality control
- remote access tools
- dashboard and monitoring tools
- configuration management systems.

In other cases, a virtual service desk allows agents to work from multiple locations, geographically dispersed. A virtual service desk requires more sophisticated supporting technology, involving more complex routing and escalation; these solutions are often cloud-based.

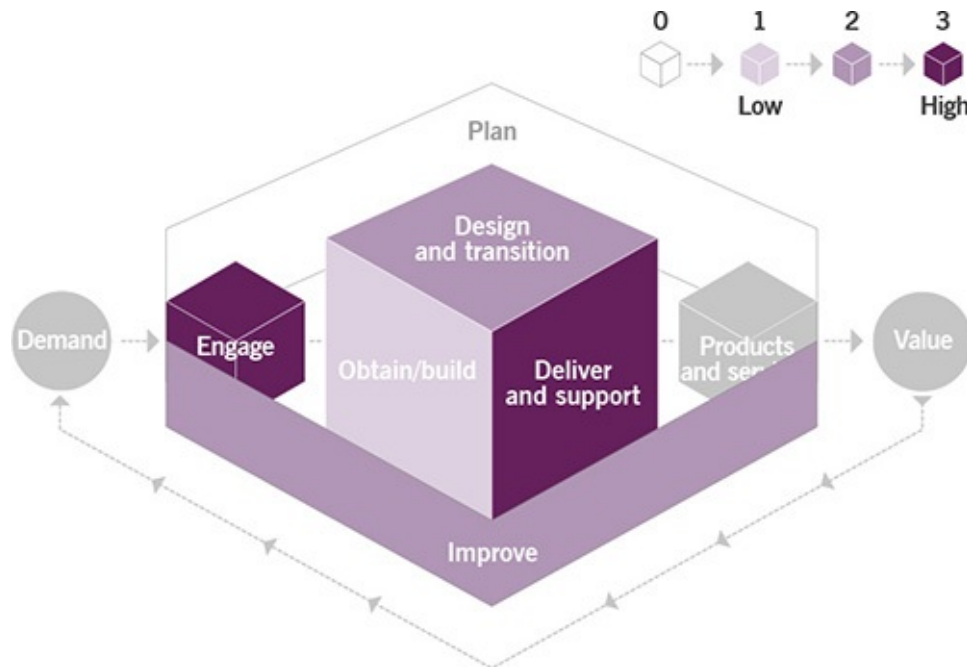


Figure 5.33 Heat map of the contribution of the service desk to value chain activities

Service desk staff require training and competency across a number of broad technical and business areas. In particular, they need to demonstrate excellent customer service skills such as empathy, incident analysis and prioritization, effective communication, and emotional intelligence. The key skill is to be able to fully understand and diagnose a specific incident in terms of business priority, and to take appropriate action to get this resolved, using available skills, knowledge, people, and processes.

Figure 5.33 shows the contribution of the service desk to the service value chain, with the practice being involved in all value chain activities except plan:

- **Improve** Service desk activities are constantly monitored and evaluated to support continual improvement, alignment, and value creation. Feedback from users is collected by the service desk to support continual improvement.
- **Engage** The service desk is the main channel for tactical and operational engagement with users.
- **Design and transition** The service desk provides a channel for communicating with users about new and changed services. Service desk staff participate in release planning, testing, and early life support.
- **Obtain/build** Service desk staff can be involved in acquiring service components used to fulfil service requests and resolve incidents.

- **Deliver and support** The service desk is the coordination point for managing incidents and service requests.

## 5.2.15 Service level management

---



### Key message

The purpose of the service level management practice is to **set clear business-based targets** for service levels, and to ensure that delivery of services is properly **assessed, monitored, and managed** against these targets.



### Definition: Service level

One or more metrics that define expected or achieved service quality.

Service level management provides the **end-to-end visibility** of the organization's services. To achieve this, service level management:

- **establishes** a **shared view of the services** and **target service levels** with customers
- **ensures** the organization meets the **defined service levels** through the collection, analysis, storage, and reporting of the relevant metrics for the identified services
- **performs** service **reviews** to ensure that the **current set of services** continues to meet the needs of the organization and its customers
- **captures** and **reports** on service **issues**, including performance against defined service levels.

The skills and competencies for service level management include relationship management, business liaison, business analysis, and commercial/supplier management. The practice requires pragmatic focus on the whole service and not simply its constituent parts; for example, simple individual metrics (such as percentage system availability) should not be taken to represent the whole service.

### 5.2.15.1 Service level agreements



### Definition: Service level agreement

A **documented agreement** between a **service provider** and a **customer** that **identifies** both services required and the expected level of service.

Service level agreements (SLAs) have long been used as a tool to measure the performance of services from the customer's point of view, and it is important that they are agreed in the wider business context. Using SLAs may present many challenges; often they do not fully reflect the wider service performance and the user experience.

Some of the **key requirements** for successful SLAs include:

- They must be related to a **defined 'service'** in the **service catalogue**; otherwise they are simply individual metrics **without a purpose**, that do not provide adequate visibility or reflect the service perspective.
- They should relate to **defined outcomes** and not simply operational metrics. This can be achieved with **balanced bundles of metrics**, such as customer satisfaction and key business outcomes.
- They should reflect an **'agreement'**, i.e. **engagement** and **discussion** between the service provider and the service consumer. It is important to **involve all stakeholders**, including partners, sponsors, users, and customers.
- They must be **simply written** and **easy to understand** and **use** for all parties.

In many cases, using single-system-based metrics as targets can result in misalignment and a disconnect between service partners regarding the success of the service delivery and the user experience. For example, if an SLA is based only on the percentage of uptime of a service, it can be deemed to be successful by the provider, yet still miss out on significant business functionalities and outcomes which are important to the consumer. This is referred to as the 'watermelon SLA' effect.

### The watermelon SLA effect

Traditional SLAs have been based on **individual activities** such as **incident resolution times**, **system availability** ('99.9'), and **volume metrics** (e.g. number of incidents or requests handled). Without a business context these metrics

are often **meaningless**. For example, although a system availability of 99.6% is impressive, this still needs to **align** with key business requirements. The system may have an acceptable unavailability of 0.4%, but if that time falls when there is an **important process happening** (such as a commercial transaction, an operating theatre in use, or point-of-sale tills in use), then customer/user satisfaction will be low, regardless of whether the SLA has been met.

This can be problematic for the service provider if it thinks it is doing a great job (the reports are all green), when in fact its customers are **dissatisfied** with the service received and also frustrated that the provider doesn't notice this. This is known as the **watermelon SLA effect**, because like a watermelon, the SLA may appear green on the outside, but is **actually red inside**.

Service level management **identifies** metrics and measures that are a truthful reflection of the **customer's actual experience** and **level of satisfaction** with the whole service. These will vary across organizations and the only way to learn what these are is to find out directly from customers.

Service level management requires focus and effort to engage and listen to the requirements, issues, concerns, and daily needs of customers:

- **Engagement** is needed to **understand** and **confirm** the actual ongoing needs and requirements of customers, not simply what is interpreted by the service provider or has been agreed several years before.
- **Listening** is important as a **relationship-building** and **trust-building** activity, to show customers that they are valued and understood. This helps to move the provider away from always being in '**solution mode**' and to build new, more constructive partnerships.

The activities of engaging and listening provide a great opportunity to build improved relationships and to focus on what really needs to be delivered. It also gives service delivery staff an experience-based understanding of the day-to-day work that is done with their technology, enabling them to deliver a more business-focused service.

Service level management involves collating and analysing information from a number of sources, including:

- **Customer engagement** This involves initial listening, discovery, and information capture on which to base metrics, measurement, and ongoing progress discussions. Consider asking customers some simple open questions such as:
  - What does your work involve?
  - How does technology help you?
  - What are your key business times, areas, people, and activities?

- What differentiates a good day from a bad day for you?
- Which of these activities is most important to you?
- What are your goals, objectives, and measurements for this year?
- What is the best measure of your success?
- On what do you base your opinion and evaluation of a service or IT/technology?
- How can we help you more?
- **Customer feedback** This is ideally gathered from a number of sources, both formal and informal, including:
  - **Surveys** These can be from immediate feedback such as follow-up questions to incidents, or from more reflective periodic surveys that gauge feedback on the overall service experience. Both are event-based.
  - **Key business-related measures** These are measures agreed between the service provider and its customer, based on what the customer values as important. This could be a bundle of SLA metrics or a very specific business activity such as a sales transaction, project completion, or operational function such as getting an ambulance to the site of an accident within x minutes.
- **Operational metrics** These are the low-level indicators of various operational activities and may include system availability, incident response and fix times, change and request processing times, and system response times.
- **Business metrics** These can be any business activity that is deemed useful or valuable by the customer and used as a means of gauging the success of the service. These can vary from some simple transactional binary measures such as ATM or POS terminal availability during business hours (09:00–17:00 daily) or successful completion of business activities such as passenger check-in.

Once this feedback is gathered and collated for ongoing review, it can be used as input to design suitable measurement and reporting models and practices.

Figure 5.34 shows the contribution of service level management to the service value chain, with the practice being applied mainly to the plan and engage activities:

- **Plan** Service level management supports planning of the product and service portfolio and service offerings with information about the actual service performance and trends.
- **Improve** Service feedback from users, as well as requirements from customers, can be a driving force for service improvement.
- **Engage** Service level management ensures ongoing engagement with customers and users through feedback processing and continual service review.
- **Design and transition** The design and development of new and changed services receives input from this practice, both through interaction with customers and as

part of the feedback loop in transition.

- **Obtain/build** Service level management provides objectives for components and service performance, as well as for measurement and reporting capabilities of the products and services.
- **Deliver and support** Service level management communicates service performance objectives to the operations and support teams and collects their feedback as an input for service improvement.

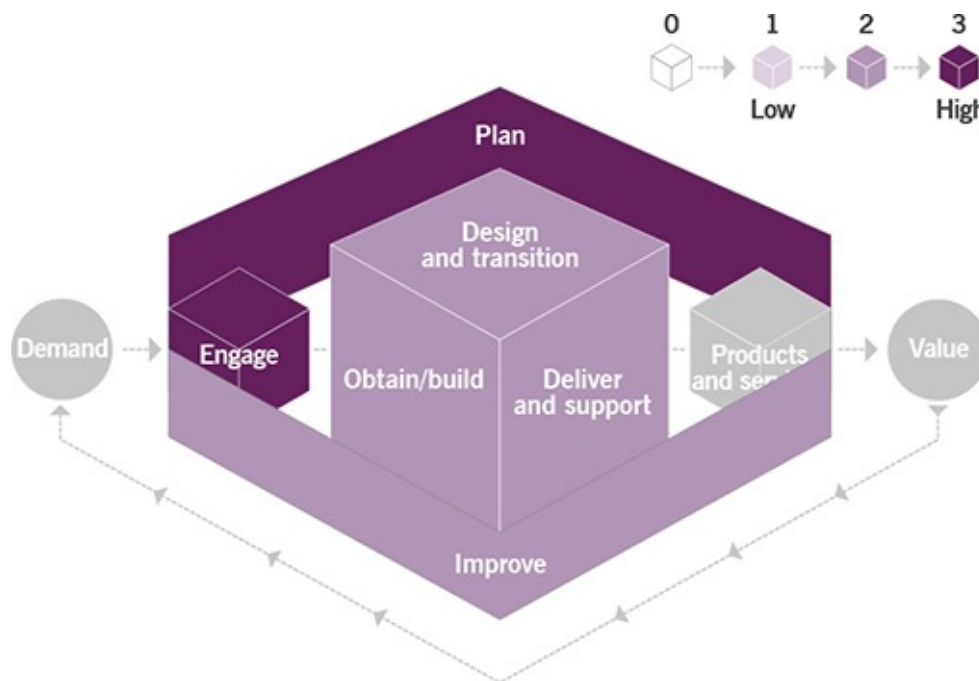


Figure 5.34 Heat map of the contribution of service level management to value chain activities

### The ITIL story: Axle's service level management



**Su:** We regularly gather feedback from our customers to analyse their requirements and needs, and update our service offerings to match their expectations.



**Radhika:** We can't put every single customer expectation into our rental agreements, but we care about all of them and do our best to meet them.



**Su:** We also monitor the quality of the services provided by our partners and suppliers, such as the work done for us by Craig's Cleaning. When doing this, we need to be sure that the quality of every part of our services meets or exceeds the expectations of our users.

## 5.2.16 Service request management

---



### Key message

The purpose of the service request management practice is to **support** the **agreed quality** of a service by **handling all pre-defined, user-initiated** service requests in an effective and user-friendly manner.



### Definition: Service request

A request from a user or a user's authorized representative that initiates a service action which has been agreed as a normal part of service delivery.

Each service request may include **one or more of the following**:

- a request for a **service delivery action** (for example, providing a report or replacing a toner cartridge)
- a request for **information** (for example, how to create a document or what the hours of the office are)
- a request for **provision of a resource or service** (for example, providing a phone or laptop to a user, or providing a virtual server for a development team)
- a request for **access to a resource or service** (for example, providing access to a file or folder)
- **feedback, compliments, and complaints** (for example, complaints about a new interface or compliments to a support team).

Fulfilment of service requests may include changes to services or their components; usually these are standard changes. Service requests are a normal part of service delivery and are not a failure or degradation of service, which are handled as incidents. Since service requests are pre-defined and pre-agreed as a normal part of service delivery, they can usually be formalized, with a clear, standard procedure for initiation, approval, fulfilment, and management. Some service requests have very simple workflows, such as a request for information. Others, such as the setup of a

new employee, may be quite complex and require contributions from many teams and systems for fulfilment. Regardless of the complexity, the steps to fulfil the request should be well-known and proven. This allows the service provider to agree times for fulfilment and to provide clear communication of the status of the request to users.

Some service requests require authorization according to financial, information security, or other policies, while others may not need any. To be handled successfully, service request management should follow these guidelines:

- Service requests and their fulfilment should be standardized and automated to the greatest degree possible.
- Policies should be established regarding what service requests will be fulfilled with limited or even no additional approvals so that fulfilment can be streamlined.
- The expectations of users regarding fulfilment times should be clearly set, based on what the organization can realistically deliver.

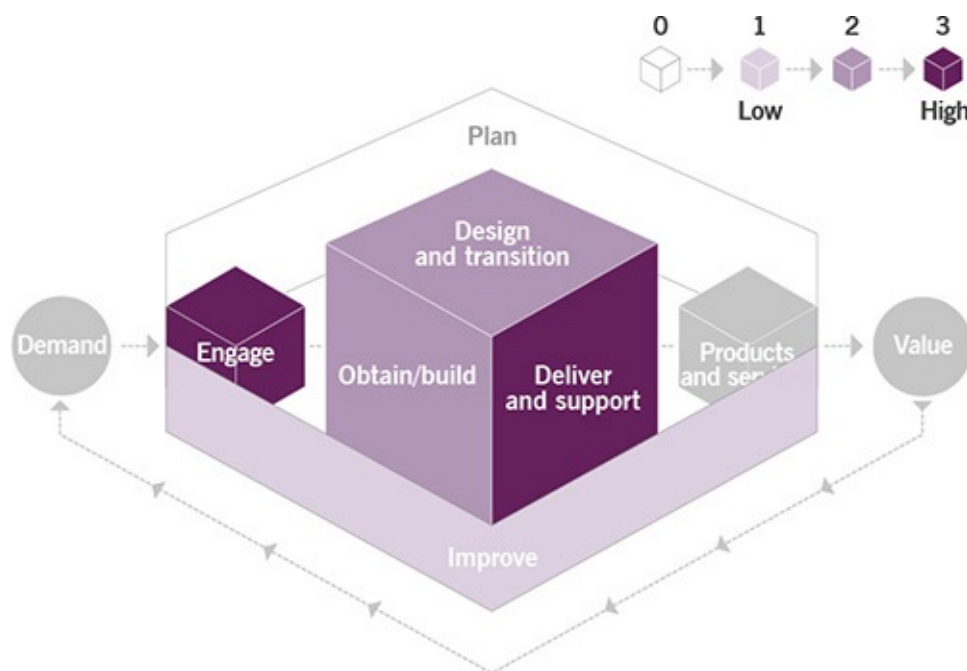


Figure 5.35 Heat map of the contribution of service request management to value chain activities

- Opportunities for improvement should be identified and implemented to produce faster fulfilment times and take advantage of automation.
- Policies and workflows should be included for the documenting and redirecting of any requests that are submitted as service requests, but which should actually be managed as incidents or changes.

Some service requests can be completely fulfilled by automation from submission to

closure, allowing for a complete self-service experience. Examples include client software installation or provision of virtual servers.

Service request management is dependent upon well-designed processes and procedures, which are operationalized through tracking and automation tools to maximize the efficiency of the practice. Different types of service request will have different fulfilment workflows, but both efficiency and maintainability will be improved if a limited number of workflow models are identified. When new service requests need to be added to the service catalogue, existing workflow models should be leveraged whenever possible.

Figure 5.35 shows the contribution of service request management to the service value chain, with the practice being involved in all service value chain activities except the plan activity:

- **Improve** Service request management can provide a channel for improvement initiatives, compliments, and complaints from users. It also contributes to improvement by providing trend, quality, and feedback information about fulfilment of requests.
- **Engage** Service request management includes regular communication to collect user-specific requirements, set expectations, and to provide status updates.
- **Design and transition** Standard service components may be transitioned to the live environment through service request fulfilment.
- **Obtain/build** Acquisition of pre-approved service components may be fulfilled through service requests.
- **Deliver and support** Service request management makes a significant contribution to normal service delivery. This activity of the value chain is mostly concerned with ensuring users continue to be productive, and sometimes depends heavily on fulfilment of their requests.

### 5.2.17 Service validation and testing

---



#### Key message

The **purpose** of the service validation and testing practice is to **ensure** that new or changed products and services **meet** defined requirements. The definition of service value is based on **input** from customers, business objectives, and regulatory requirements, and is **documented** as part of the value chain activity of design and transition. These inputs are used to **establish measurable quality**

and performance indicators that support the definition of assurance criteria and testing requirements.

### 5.2.17.1 Service validation

---

Service validation focuses on establishing deployment and release management acceptance criteria (conditions that must be met for production readiness), which are verified through testing. Acceptance criteria can be either utility- or warranty-focused, and are defined through understanding customer, regulatory, business, risk management, and security requirements.

The service validation activities of this practice establish, verify, and document both utility- and warranty-focused service assurance criteria and form the basis for the scope and focus of testing activities.

### 5.2.17.2 Testing

---

A test strategy defines an overall approach to testing. It can apply to an environment, a platform, a set of services, or an individual service. Testing should be carried out equally on both in-house developed systems and externally developed solutions. The test strategy is based on the service acceptance criteria, and should align with the requirements of appropriate stakeholders to ensure testing matches the risk appetite and is fit for purpose.

Typical test types include:

- **Utility/functional tests:**
    - **Unit test** A test of a single system component
    - **System test** Overall testing of the system, including software and platforms
    - **Integration test** Testing a group of dependent software modules together
  - **Regression test** Testing whether previously working functions were impacted.
- Warranty/non-functional tests:
- **Performance and capacity test** Checking speed and capacity under load
  - **Security test** Testing vulnerability, policy compliance, penetration, and denial of service risk
  - **Compliance test** Checking that legal and regulatory requirements have been met
  - **Operational test** Testing for backup, event monitoring, failover, recovery, and reporting
  - **Warranty requirements test** Checking for verification of necessary documentation, training, support model definition, and knowledge transfer



Figure 5.36 Heat map of the contribution of service validation and testing to value chain activities

## 5.3 Technical management practices

### 5.3.1 Deployment management



#### Key message

The purpose of the deployment management practice is to **move new or changed** hardware, software, documentation, processes, or any other component to **live environments**. It may also be involved in deploying components to **other environments** for testing or staging.

Deployment management works closely with release management and change control, but is a separate practice. In some organizations the term ‘provisioning’ is used to describe the deployment of infrastructure, and deployment is only used to mean software deployment, but in this case the term deployment is used to mean both.

There are a number of distinct approaches that can be used for deployment. Many organizations use a combination of these approaches, depending on their specific services and requirements as well as the release sizes, types, and impact.

- **Phased deployment** The new or changed components are deployed to just part of the production environment at a time, for example to users in one office, or one country. This operation is repeated as many times as needed until the deployment is complete.
- **Continuous delivery** Components are integrated, tested, and deployed when they are needed, providing frequent opportunities for customer feedback loops.
- **Big bang deployment** New or changed components are deployed to all targets at the same time. This approach is sometimes needed when dependencies prevent the simultaneous use of both the old and new components. For example, there could be a database schema change that is not compatible with previous versions of some components.
- **Pull deployment** New or changed software is made available in a controlled repository, and users download the software to client devices when they choose.

This allows users to control the timing of updates, and can be integrated with service request management to enable users to request software only when it is needed.

Components that are available for deployment should be maintained in one or more secure locations to ensure that they are not modified before deployment. These locations are collectively referred to as a definitive media library for software and documentation, and a definitive hardware store for hardware components.

Tools that support deployment are many and varied. They are often integrated with configuration management tools, and can provide support for audit and change management. Most organizations have tools for deploying client software, and these may be integrated with a service portal to support a request management practice.

Communication around deployments is part of release management. Individual deployments are not generally of interest to users and customers until they are released.

If infrastructure is provided as a service, then deployment of new or changed servers, storage, or networking is typically managed by the organization, often treating the infrastructure as a code, so that the organization can automate deployment. In these environments it is possible that some deployments may be under the control of the supplier, such as the installation of firmware updates, or if they provide the operating system as well as the infrastructure they may deploy operating system patches. The IT organization must ensure that they know what deployments are planned, and which have happened, to maintain a controlled environment.

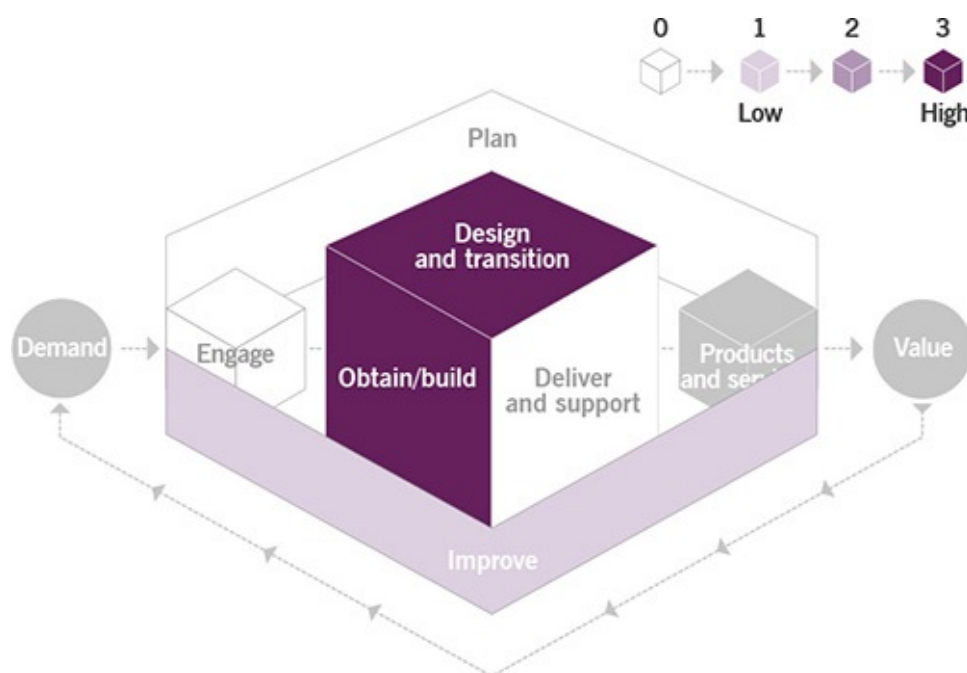


Figure 5.37 Heat map of the contribution of deployment management to value chain activities

If application development is provided as a service, then deployment may be carried out by the external application developer, by the in-house IT department, or by a service integrator. Again, it is essential that the organization is aware of all deployments so that a controlled environment can be maintained.

In an environment with multiple suppliers it is important to understand the scope and boundaries of each organization's deployment activities, and how these will interact. Most organizations have a process for deployment, and this is often supported with standard tools and detailed procedures to ensure that software is deployed in a consistent way. It is common to have different processes for different environments. For example, there may be one process for the deployment of client application software, and a completely different process for the deployment of server operating system patches.

Figure 5.37 shows the contribution of deployment management to the service value chain, with the practice being applied mainly to the design and transition, and obtain/build value chain activities, but also to the improve activity:

- **Improve** Some improvements may require components to be deployed before they can be delivered, and these should be planned and managed in the same way as any other deployment.
- **Design and transition** Deployment management moves new and changed components to live environments, so it is a vital element of this value chain activity.
- **Obtain/build** Changes can be deployed incrementally as part of this value chain activity. This is especially common in DevOps environments using a complete automated toolchain for continuous integration, delivery, and deployment.

### The ITIL story: Axle's deployment management



**Marco:** Before we deploy changes to our booking app, we release the changes to a test environment. After thorough testing, we make the changes available to our users and customers.



**Radhika:** We recently realized that the same logic can be applied to some of our non-digital services and components. For example, last month we introduced two brand new hybrid models for hire in some bigger cities. We created a promotional service offering for the new cars, updated our marketing materials, trained our technicians to work with the new models, and deployed everything in advance – including the vehicles. This happened before the official launch of the hybrid cars by the manufacturer. And of course, it happened with their permission.



**Su:** *By the time the launch date arrived, we were ready to go. We made the cars available to hire that very day.*



**Henri:** *Partnering with our manufacturer meant we had a successful and well-prepared launch that created a buzz with our customers and with theirs.*

### 5.3.2 Infrastructure and platform management



#### Key message

The purpose of the infrastructure and platform management practice is to **oversee** the infrastructure and platforms used by an organization. When carried out properly, this practice enables the **monitoring** of technology solutions available to the organization, including the technology of external service providers.

IT infrastructure is the physical and/or virtual technology resources, such as servers, storage, networks, client hardware, middleware, and operating systems software, that provide the environments needed to deliver IT services. This includes any CI a customer uses to access the service or consume a product. IT infrastructure may be managed by the service provider or by an external supplier as dedicated, shared, or cloud services. Infrastructure and platform management may also include the buildings and facilities an organization uses to run its IT infrastructure.

The infrastructure and platform management practice includes the provision of technology needed to support activities that create value for the organization and its stakeholders. This can include being ready to adopt new technologies such as machine learning, chatbots, artificial intelligence, mobile device management, and enterprise mobility management.

It is important to consider that every single organization must develop its own strategy to achieve the intended outcome with any type of infrastructure or platform. Each organization should design its own cloud management system to orchestrate all the interrelated components of infrastructure and platform with its business goals and the intended service quality and operational efficiency.

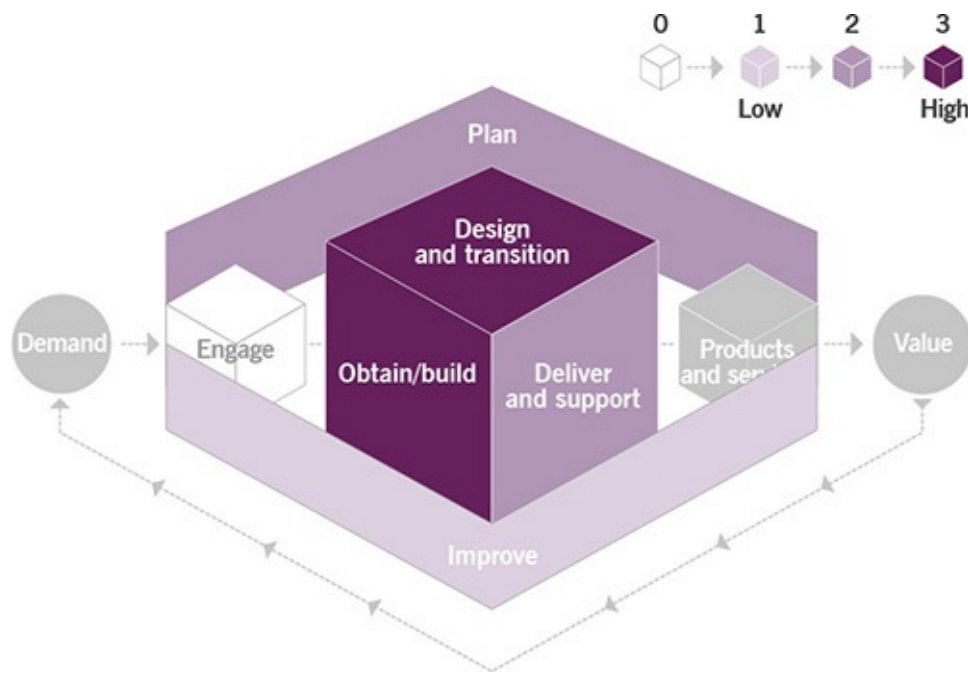


Figure 5.38 Heat map of the contribution of infrastructure and platform management to value chain activities

Figure 5.38 shows the contribution of infrastructure and platform management to the service value chain, with the practice being involved in all value chain activities except the engage activity:

- **Plan** Infrastructure and platform management provides information about technology opportunities and constraints that is used for the organization's strategic and tactical planning.
- **Improve** Information about technology opportunities that can support continual improvement, and any constraints of the technologies in use, is provided by this practice.
- **Design and transition** Product and service design benefits from the information provided about technology opportunities and constraints.
- **Obtain/build** Infrastructure and platform management is a critical contributor to this activity as it provides necessary information about the components to be obtained.
- **Deliver and support** At the operational level, infrastructure and platform management supports ongoing maintenance of the services and the infrastructure, including any executions of patch management, backups, etc.

## Cloud service models

Cloud service models include:

- **Software as a service** (SaaS) The consumer can use the **applications** running

in the cloud infrastructure without having to control or even manage the underlying cloud infrastructure.

- Platform as a service (PaaS) The consumer can deploy onto the cloud acquired applications created using programming languages, services, libraries, and/or tools supported by the supplier without having to control or even manage the underlying **cloud infrastructure**. They have control over the deployed applications and sometimes the configuration settings for the application and hosting environment.
- Infrastructure as a service (IaaS) The consumer can get processing, storage, and/or any other **computing resources** without having to control the underlying infrastructure.

## Cloud service deployment models

Every service model can be deployed in several ways, either independently or using a mix of the following:

- Private cloud This type of cloud may be **located** within the organization's premises or outside of it. It is a cloud infrastructure or platform to be used exclusively by a specific organization which, at the same time, can have one or several consumers. This cloud is normally managed and owned by an organization, a provider, or a combination of both.
- Public cloud This type of cloud is located on the **cloud provider premises**. It is provisioned for open use and may be owned, managed, and operated by any type of organization interested in using it.
- Community cloud A community cloud may be owned, managed, and operated by one or more of the **stakeholders in the community**, and it may exist on or off the organization's premises. This cloud deployment model consists of several cloud services that are meant to support and share a collection of cloud service customers with the same requirements and who have a relationship with one another.
- Hybrid cloud This cloud infrastructure is a **composition of two or more** distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability.

## ITIL practices and cloud computing

The advent of the cloud has been one of the greatest challenges and opportunities within the IT world for decades. The promise of rapid, elastic

storage and IT services available at the touch of a button is one that many organizations struggle to deliver internally, not because the benefits are not there to be had, but rather because their own ITSM processes and controls have not been adapted to support a radically different way of working.

The management and control of IT services is a **key skill of IT departments** no matter where those services are physically located, and the processes and controls offered by ITIL are readily adaptable to support the management of those cloud services.

A **coordinated response** to the management of cloud services is essential. Organizations that attempt to address only a cloud service provision as an operational issue will suffer on the tactical front, just as an organization that attempts to control cloud services on a tactical front will suffer at the strategic level. A joined-up approach covering all three levels, strategic, tactical, and operational, is required.

Apart from the infrastructure and platform management practice, the operation and management of cloud-based services involves many other practices. It should be noted that this is not a comprehensive list:

- **Service financial management** One of the adjustments that IT departments often have to make for cloud computing is to their **fiscal planning**, which typically uses both traditional capital expenditure (**CAPEX**) and operational expenditure (**OPEX**). With the advent of cloud computing, OPEX is preferred over CAPEX, as cloud services are often consumed as utilities and paid out of the operational budget. If cloud services are quicker and easier to access than in-house services, the costs associated with them will grow as more parts of the business use them. The IT cost model must be adjusted, and the service financial management practice can help to determine the techniques and controls required to ensure that the organization does not run out of OPEX unexpectedly.
- **Supplier management** The focus of this practice will need to change from simply selecting suppliers and onboarding them to acting as the front end for a full-on release management process. This will ensure that areas such as IT security, data protection, and regulatory compliance are routinely assessed prior to the onboarding of a new cloud offering.
- **Capacity and performance management** Coupled with service financial management, this practice should **establish** and **monitor** budgets, with **thresholds** tracked and warnings published if an upswing in demand leads to an increase in the cost of cloud services.
- **Change control** The boundaries of this practice will have to be redefined, as cloud service providers often make changes with minimal customer involvement, and almost no customer approval. Products and services built on top of cloud services will need to make far greater use of standard

changes to unlock the benefits that cloud platforms (and associated business models) provide.

- **Incident management** The focus of this practice will change from **knowing** how to fix in-house issues, to **knowing** which service is supported by which cloud provider, and **what information** they will require to resolve an issue. **Greater care** will be needed to support impacted customers and teams.
- **Deployment management** This practice will continue to be **critical** to IT departments, but the ability to safely onboard or offboard a cloud provider will become a common requirement for IT departments. Deployment management will be a **key capability** for successful IT organizations, to ensure new cloud capabilities are rapidly deployed and embedded within the in-house service offerings.

### 5.3.3 Software development and management

---



#### Key message

The purpose of the software development and management practice is to ensure that applications meet **internal and external stakeholder needs**, in terms of functionality, reliability, maintainability, compliance, and auditability.

The term ‘software’ can be used to describe anything from a single program (or suite of programs) to larger constructs (such as an operating system, an operating environment, or a database) on which various smaller application programs, processes, or workflows can run. Therefore the term includes, but is not limited to, desktop applications, or mobile apps, embedded software (controlling machines and devices), and websites.

Software applications, whether developed in house or by a partner or vendor, are of critical importance in the delivery of customer value in technology-enabled business services. As a result, software development and management is a key practice in every modern IT organization, ensuring that applications are fit for purpose and use.

The software development and management practice encompasses **activities** such as:

- solution **architecture**
- solution **design** (user interface, CX, service design, etc.)

- software **development**
- software **testing** (which can include several components, such as unit testing, integration testing, regression testing, information security testing, and user acceptance testing)
- **management** of **code repositories** or **libraries** to maintain integrity of artefacts
- **package creation**, for the effective and efficient deployment of the application
- **version control, sharing, and ongoing management** of smaller blocks of code.

The two generally accepted approaches to software development are referred to as the waterfall and Agile methods (see section 5.1.8 for more information on these methods).

Software management is a wider practice, encompassing the ongoing activities of designing, testing, operating, and improving software applications so they continue to facilitate value creation. Software components can be continually evaluated using a lifecycle that tracks the component from ideation through to ongoing improvement, and eventually retirement. This lifecycle is represented in Figure 5.39.

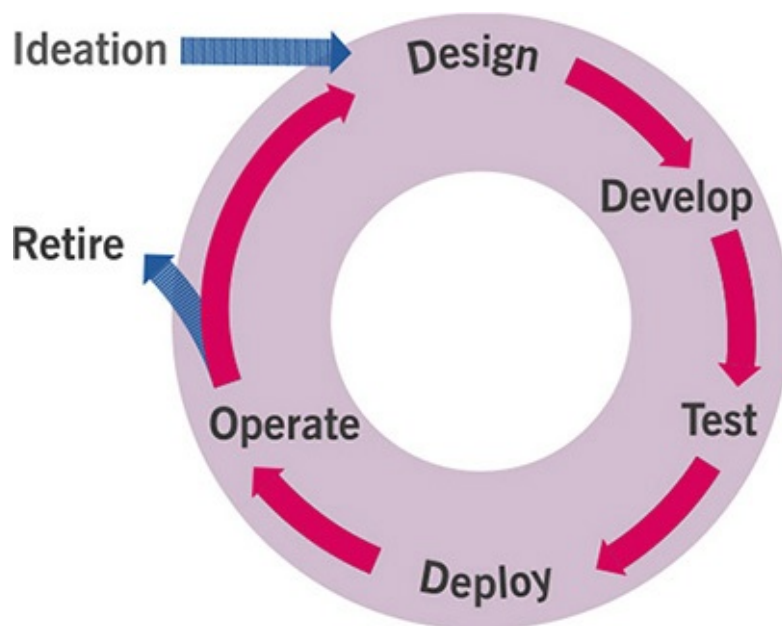


Figure 5.39 The software lifecycle

Figure 5.40 shows the contribution of software development and management to the service value chain, with the practice being involved in all value chain activities except the engage activity:

- **Plan** Software development and management provides information about opportunities and constraints related to the creation and changing of the organization's software.
- **Improve** Service improvements involving software components of the services, especially those developed in house, rely on this practice.

- **Design and transition** Software development and management allows the organization to holistically design and manage changes to products and services.
- **Obtain/build** The creation of in-house products and the configuration of products developed by partners and suppliers depend on this practice.
- **Deliver and support** Software development and management provides delivery and support teams with documentation needed to use products that facilitate the co-creation of value.

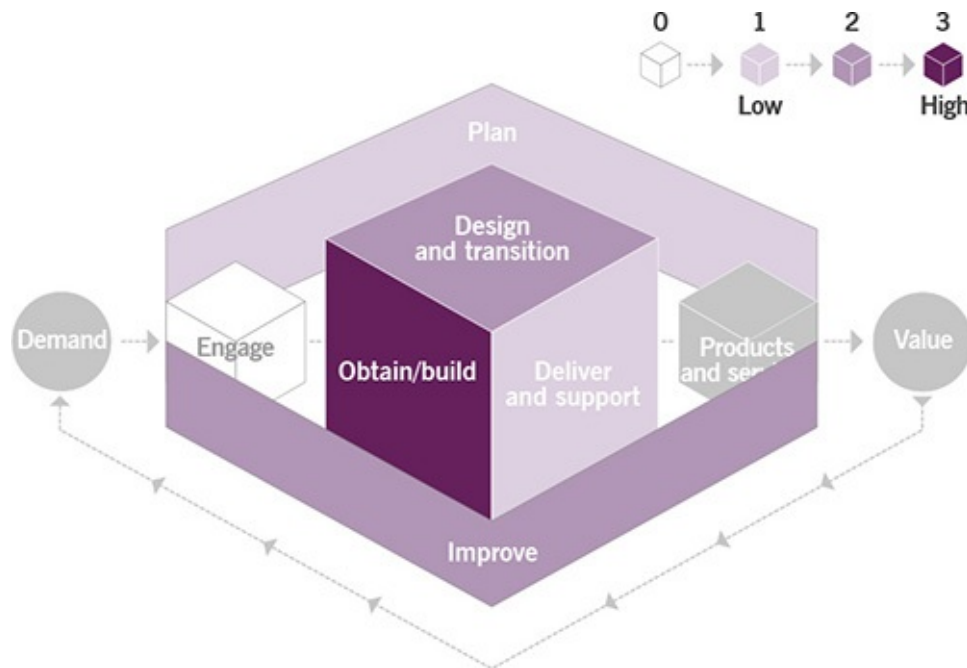


Figure 5.40 Heat map of the contribution of software development and management to value chain activities